Proxy 서버를 이용하는 효율적인 키 분배 프로토콜

양 형 규[†]

요 약

키 분배 프로토콜은 안전한 암호 시스템을 구성하기 위한 필수적인 요소이며, 키 분배 프로토콜에 대한 연구·개발은 꾸준히 이루어지고 있다. 공개키 암호 방식을 이용하는 대부분의 키 분배 프로토콜은 Diffie-Hellman 방식에 기반하므로 세션키 설정에 요구되는 계산량이 많아 무선 인터넷 환경에는 적용하기 어렵다. 본 논문에서는 proxy 서버를 이용하여 무선 단말기 사용자에게 요구되는 계산량을 줄일 수 있는 개선된 키 분배 프로토콜을 제안한다. 제안하는 방식은 두 사용자간에 키 분배가 이루어지는 기존의 키 분배 프로토콜의 안전성을 그대로 유지하면서 단말 사용자에게 요구되는 계산량을 줄일 수 있다는 장점이 있다.

Efficient Key Agreement Protocols using Proxy Server

Hyung-Kyu Yang[†]

ABSTRACT

A key agreement protocol is the most important part to establish a secure cryptographic system and the effort to standardize the key agreement protocols is in rapid progress. Several efficient and secure key agreement protocols have been proposed so far since Diffie-Hellman proposed a public key agreement system in 1976. But, since Diffie-Hellman based key agreement protocols need a lot of computation to establish the session key, they are not suitable for wireless Internet environment. In this paper, we propose the efficient key agreement protocol using proxy server. The proposed protocol gives the security equivalent to that the Diffie-Hellman based protocol and the computation work of mobile user can be decreased using proxy server.

키워드 : 키동의 프로토콜(Key Agreement Protocol), 무선 인터넷(Wireless Internet), Diffie-Hellman Primitive, Proxy 서버 기반 암호 시스템(*Cryptosystem based on Proxy Server*)

1. 서 론

최근 들어 휴대폰이나 PDA와 같은 무선 통신용 단말기를 이용한 무선 인터넷 서비스가 활성화되고 있다. 따라서, 사용자들은 어디서든지 자신의 휴대용 단말기를 이용하여 웹 서버에 접속하고 이를 통해 인터넷 뱅킹, 온라인 쇼핑, 주식 거래와 같은 다양한 서비스를 이용할 수 있게 되었다.

또한, 보다 안전한 무선 인터넷 서비스를 제공하기 위해서는 유선 환경에서와 같이 암호 시스템의 사용이 필수적으로 요구되며, 안전한 암호 시스템의 사용에 있어 키 분배 프로토콜은 필수적인 구성 요소이다.

그러나 무선 인터넷 환경은 유선 환경에 비해 사용하는 단말기의 계산 능력이나 전원 장치가 부족하므로, 유선 환경과 같이 공개키 암호 방식을 적용하는데 많은 어려움이 있다. 따라서 최근 들어 이러한 문제점을 해결하기 위한 여러 가지 연구가 진행되고 있으며, 대표적인 결과로는 타원곡선을 이용한 암호 시스템이나 proxy 서버를 이용하는 방식 등이 있다.

Proxy 서버를 이용하는 방식은 단말 사용자에게 요구되는 많은 양의 계산을 사전에 선택한 임의의 서버에게 의존하는 방식으로 디지털 서명이나 인증 시스템 등에서 사용되고 있다.

본 논문에서는 기존에 제안된 Diffie-Hellman의 키 분배 프로토콜을 변형하여 무선 인터넷 환경에 적합한 프로토콜을 제안하고자 한다. 제안하는 프로토콜은 기존의 프로토콜이 가지고 있던 안전성은 그대로 만족하면서 proxy 서버를 이용하여 단말 사용자에게 요구되는 계산량을 감소시킬 수 있는 방식이다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 Diffie-Hellman에 기반한 키 분배 프로토콜 중 ANSI X9.42[1]에서 제안하는 dhHybridOneFlow 방식에 대해 간단히 설명한다. 3장에서는 2장에서 설명한 프로토콜을 변형하여, 단말 사용자의 계산량을 감소시킨 2개의 proxy 서버 기반 프로토콜을 제안하고 제안하는 프로토콜의 특징을 분석한다. 다음으로 4장에서는 제안하는 프로토콜의 안전성을 분석하고 마지막으로 5장에서 결론을 맺는다.

† 정 회 원 : 강남대학교 컴퓨터미디어공학부 교수
논문접수 : 2004년 9월 24일, 심사완료 : 2004년 11월 23일

2. 연구 배경

키 분배 프로토콜(key distribution protocol)이란 안전하지 않은 통신로를 이용하여 두 사용자 간에 안전하고 효율적인 비밀 세션키(session key)를 공유하기 위한 메커니즘이다. 이러한 키 분배 프로토콜은 1976년 Diffie-Hellman[4]에 의해 처음으로 제안된 이후, 많은 연구가 진행되어 현재까지 다양한 프로토콜들이 제안되었다[3, 5].

키 분배 프로토콜은 세션키를 설정하는 유형에 따라 크게 두 가지로 나눌 수 있다. 먼저, 사용자 A와 사용자 B가 공유하고자 하는 비밀 세션키를 어느 누구도 미리 결정하지 않고 두 사용자간의 합의에 의해 설정하는 키 동의(key agreement) 방식과 사용자 A가 비밀 세션키를 일방적으로 선택하여 세션키를 공유하고자 하는 사용자 B에게 안전하게 전송하는 키 전송(key transport)방식이 있다[9].

또한, 키 분배 방식은 세션키 설정을 위해 사용하는 암호 방식에 따라, 관용 암호 시스템을 이용한 방식과 공개키 암호 시스템을 이용하는 방식으로 나눌 수 있다. 관용 암호 시스템을 이용하기 위해서는 사전에 안전한 통신로를 이용하여 두 사용자간에 미리 비밀키를 공유해야하므로 개방된 통신로를 이용하는 컴퓨터 네트워크에 적용하는 데에는 어려움이 많다. 따라서, 공개키 암호 시스템을 이용한 키 분배 방식이 많이 사용되고 있다.

지금까지 공개키 암호 방식을 이용하는 많은 키 분배 프로토콜이 제안되었으며, 최근 들어 이러한 키 분배 프로토콜에 대한 표준화 작업이 진행되고 있다. 키 분배 프로토콜에 대한 대표적인 표준에는 ANSI X9.42/X9.63, IEEE P1363[6], PKCS#3 등이 있다.

본 절에서는 X9.42에서 제안하고 있는 Diffie-Hellman 기반 키 분배 프로토콜 중 dhHybridOneFlow 방식에 대해 간단히 설명하고, 다음 절에서 이들 방식을 변형한 새로운 프로토콜을 제안하고자 한다.

먼저, 프로토콜의 설명에 사용하는 기호의 정의는 다음과 같다. p는 512 비트 이상의 큰 소수이고 g는 위수가 q인 Z_p상의 원소이다. 그리고 x_i는 사용자 i의 고정된(static) 비밀키이고 $y_i = g^{x_i} \bmod p$는 사용자 i의 고정된 공개키이며, $\|$는 연접(concatenation)을 의미한다.

[dhHybridOneFlow 프로토콜]

① 사용자 A는 랜덤 수 $r_A \in_R Z_q$를 선택하고 $t_A \equiv g^{r_A} \bmod p$를 계산한다.

② 사용자 A는 t_A와 자신의 고정된 공개키 y_A를 사용자 B에게 전송한다.

③ 사용자 A는 다음과 같이 공유 비밀정보 C를 계산한다.

$$C \equiv y_B^{x_A} \| y_B^{r_A} \equiv g^{x_A x_B} \| g^{r_A x_B} \bmod p$$

④ 사용자 B는 다음과 같이 공유 비밀정보 C를 계산한다.

$$C \equiv y_A^{x_B} \| t_A^{x_B} \equiv g^{x_A x_B} \| g^{r_A x_B} \bmod p$$

이 방식은 세션키 설정에 필요한 통신 회수는 1번이며, 양방향 묵시적 키 인증과 일방향 key freshness를 제공한다. 그리고 세션키를 설정하기 위해 사용자 A는 3번, 사용자 B는 2번의 모듈라 멱승이 필요하다.

3. 제안하는 프로토콜

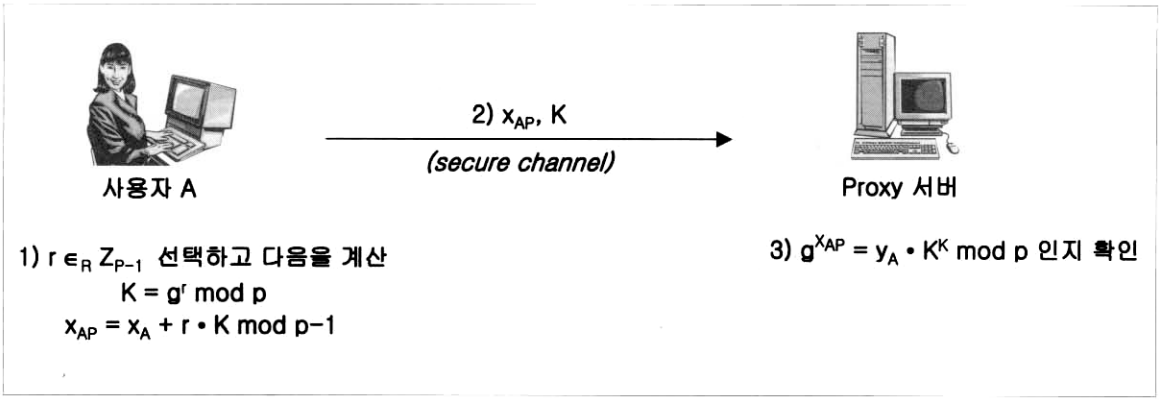
본 절에서는 proxy 서버를 이용하여 단말 사용자에게 요구되는 계산량을 감소시킬 수 있는 변형된 프로토콜을 제안한다. 제안하는 프로토콜은 ANSI X9.42[1]의 dhHybrid-OneFlow 프로토콜을 변형한 방식으로, 제안하는 프로토콜 1은 웹 서버가 proxy 서버의 신원을 명시적으로 인증할 수 있는 방식이며, 제안하는 프로토콜 2는 웹 서버가 proxy 서버의 신원을 묵시적으로 인증할 수 있는 방식이다.

제안하는 방식 1은 proxy 서버가 계산한 정보에 디지털 서명을 생성하여 전송하므로, 웹 서버는 proxy 서버의 신원과 수신한 정보의 정당성에 대해 명시적으로 인증할 수 있다는 장점이 있다. 그러나 웹 서버는 세션키를 계산하기 전에 전송 정보의 정당성을 확인하기 위해 디지털 서명을 검증해야 하므로 웹 서버의 계산량이 기존의 dhHybrid-OneFlow 프로토콜에 비해 증가한다는 단점이 있다. 이를 해결하기 위해, 제안하는 방식 2에서는 디지털 서명을 사용하지 않고 웹 서버가 proxy 서버로부터 수신한 정보에 대해 묵시적으로 인증할 수 있도록 개선하였다. 제안하는 방식 (2)에서 웹 서버는 사용자의 위임을 받은 정당한 proxy 서버 외에 다른 사용자는 해당 정보를 생성할 수 없음을 확인할 수 있으므로, 수신한 정보에 대해 묵시적으로 인증할 수 있다. 또한, 디지털 서명 대신 exclusive-or 연산을 수행하므로 웹 서버에게 요구되는 계산량을 증가시키지 않는다는 장점이 있다.

3.1 시스템 파라미터

먼저, 제안하는 프로토콜에서 사용하는 시스템 파라미터의 정의는 다음과 같다.

- p : 512 비트 이상의 큰 소수
- g : Z_p상의 원시 원소 ($g^{p-1} \equiv \bmod p$)
- x_A : 사용자 A의 비밀키, $x_A \in Z_{p-1}$
- y_A : 사용자 A의 공개키, $y_A = g^{x_A} \bmod p$
- x_{AP} : 사용자 A가 proxy 서버에게 전달하는 비밀 위임 정보
- y_{AP} : procy 서버가 사용자 A의 위임을 받았음을 확인하는데 사용하는 공개 정보, $y_{AP} = g^{x_{AP}} \bmod p$
- x_B : 웹 서버 B의 비밀키, $x_B \in Z_{p-1}$
- y_B : 웹 서버 B의 공개키, $y_B = g^{x_B} \bmod p$



(그림 1) Proxy 키 생성 과정

- K_{AP} : 사용자 A와 proxy 서버사이에 사전에 공유한 비밀키
- $E(\,)/D(\,)$: 대칭키 암호 방식의 암호화/복호화 알고리즘

3.2 세션키 설정 과정

제안하는 proxy 서버를 이용하는 키 분배 프로토콜은 proxy 서버에게 proxy 키를 생성하여 위임하는 단계와 사용자와 웹 서버 사이의 비밀 정보 공유 단계, 그리고 키 유도 함수를 이용하여 실제 세션키를 계산하는 단계로 나눌 수 있다.

먼저, 세션키 설정 과정은 이동 단말기를 가진 사용자는 등록 센터와 같이 proxy 서버를 선택하고 위임 정보를 생성하여 단말기에 저장할 수 있는 곳에 직접 방문하는 것으로, 사용자는 처음에 proxy 서버에 가입할 때 proxy 키 생성 단계를 한번만 수행하고 해당 정보를 proxy 서버에 저장하게 된다. proxy 키 위임 과정은 M. Mambo 등이 [7-8]에서 사용한 방식을 이용하며 자세한 과정은 다음과 같다.

[Proxy 키 생성 단계](그림 1 참조)

① 사용자 A는 다음과 같이 위임 정보를 생성하여 Proxy 서버에게 (x_{AP}, K)를 안전하게 전송한다.

$$r \in_R Z_{p-1} - \{0\}$$
$$K \equiv g^r \bmod p$$
$$x_{AP} = x_A + r \cdot K \bmod p-1$$

② Proxy 서버는 다음 식을 이용하여 위임 정보의 정당성을 확인한다.

$$g^{x_{AP}} = y_A \cdot K^K \bmod p$$

제안하는 방식에서 proxy 서버의 도움을 받아 사용자 A와 웹 서버 B 사이에 비밀 정보를 공유하는 세션키 생성 과정은 다음과 같다.

[제안하는 키 분배 프로토콜 1]

① 웹 서버 B와의 비밀 통신을 위해 무선 단말기를 사용하는 사용자 A는 자신의 proxy 서버에게 세션키 설정을 요청한다.

② Proxy 서버는 $r_p \in_R Z_{p-1}$ 선택하고 $t_p \equiv g^{r_p} \bmod p$를 계산한 후, 다음과 같이 디지털 서명을 생성한다.

$$r = H(g^x \bmod p, t_p) \text{ (단, } x \in_R [1, p-1])$$
$$s = x/(r + x_{AP}) \bmod p-1$$

③ proxy 서버는 (y_{AP}, K, t_p, r, s)를 웹 서버 B에게 전송한다.

④ 웹 서버 B는 $y_{AP} \equiv y_A \cdot K^K \bmod p$인지 확인하여, proxy 서버가 사용자 A의 요청에 의해 세션키 설정을 요구하였음을 확인한다.

⑤ 웹 서버 B는 $r' \equiv (y_{AP} \cdot g^r)^s \bmod p$를 계산한 후, $r = H(r', t_p)$인지 확인한다.

⑥ 웹 서버 B는 다음과 같이 세션키를 계산하기 위한 비밀 공유 정보 C를 계산한다.

$$C \equiv y_A^{x_B} \| t_p^{x_B} \equiv g^{x_A x_B} \| g^{r_p x_B} \bmod p$$

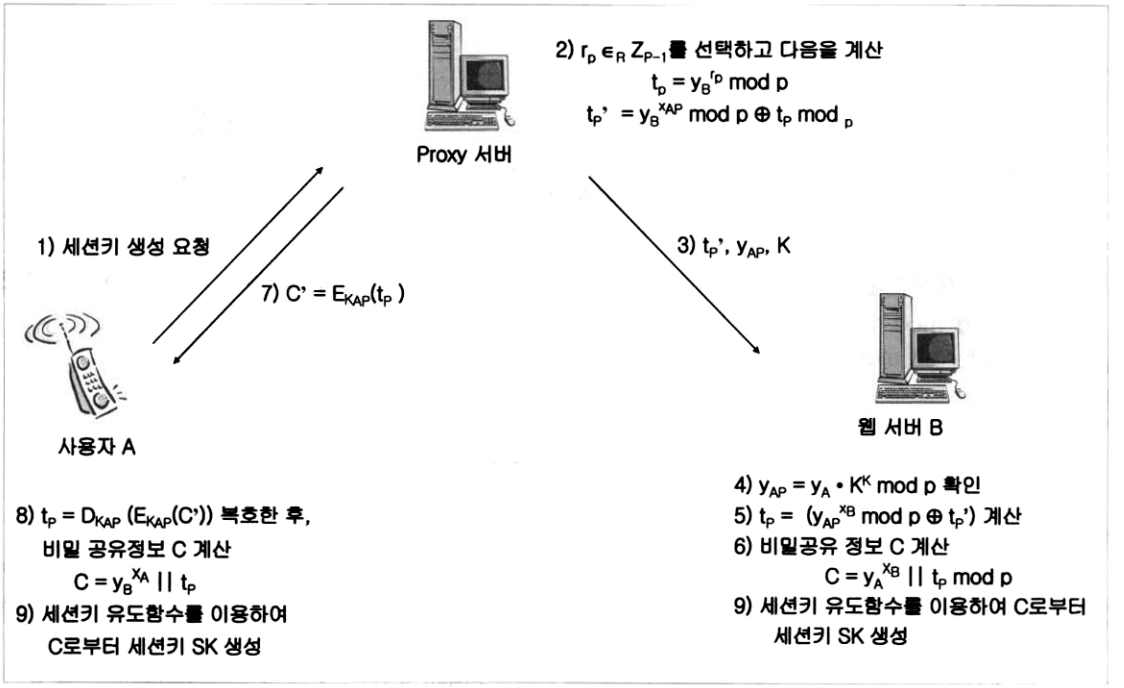
⑦ Proxy 서버는 $C' \equiv y_B^{r_p} \bmod p$를 계산하여 사전에 공유한 비밀키 K_{AP}를 이용하여 암호화한 후, $E_{K_{AP}}(C')$를 사용자 A에게 전송한다.

⑧ 사용자 A는 사전에 공유한 비밀키 K_{AP}를 이용하여 C'를 복호하고, 다음과 같이 C를 생성한다.

$$C = (y_B)^{x_A} \| C'$$

단, 여기서 $y_B^{x_A}$는 항상 일정한 값이므로, 사전 계산이 가능하다.

⑨ 사용자 A와 웹 서버 B는 각각 키 유도 함수를 이용하여 실제 세션키를 계산한다.



(그림 2) 제안하는 키 분배 프로토콜 2

[제안하는 키 분배 프로토콜 2]

① 웹 서버 B와의 비밀 통신을 위해 무선 단말기를 사용하는 사용자 A는 자신의 proxy 서버에게 세션키 설정을 요청한다.

② Proxy 서버는 $r_p \in_R Z_{p-1}$를 선택하고, 다음과 같이 t_p, t_p'를 계산한다.

$$t_p \equiv y_B^{r_p} \bmod p$$

$$t_p' \equiv (y_B^{X_{AP}} \bmod p) \oplus t_p$$

③ Proxy 서버는 (y_{AP}, K, t_p')를 웹 서버 B에게 전송한다.

④ 웹 서버 B는 다음 식을 이용하여 proxy 서버가 사용자 A의 요청에 의해 세션키 설정을 요구하였음을 확인한다.

$$y_{AP} \equiv y_A \cdot K^K \bmod p$$

⑤ 웹 서버 B는 다음과 같이 t_p 값을 계산한다.

$$t_p \equiv (y_{AP}^{X_B} \bmod p) \oplus t_p'$$

⑥ 웹 서버 B는 다음과 같이 세션키를 계산하기 위한 비밀 공유 정보 C를 계산한다.

$$C \equiv y_A^{X_B} \| t_p \equiv g^{X_A X_B} \| g^{r_p X_B} \bmod p$$

⑦ Proxy 서버는 ②에서 계산한 t_p를 사전에 공유한 비밀키 K_{AP}를 이용하여 암호화한 후, 사용자 A에게 $C' = E_{K_{AP}}(t_p)$를 전송한다.

⑧ 사용자 A는 사전에 공유한 비밀키 K_{AP}를 이용하여 C'를 복호하고, 다음과 같이 C를 생성한다.

$$C = (y_B)^{X_A} \| t_p \equiv g^{X_A X_B} \| g^{r_p X_B} \bmod p$$

단, 여기서 $y_B^{X_A}$는 항상 일정한 값이므로, 사전 계산이 가능하다.

⑨ 사용자 A와 웹 서버 B는 각각 세션키 유도 함수를 이용하여 실제 세션키를 계산한다.

마지막으로, 앞의 과정을 통해 사용자와 웹 서버 사이에 공유된 비밀 정보 C를 이용하여 암호 통신을 수행하기 위해, 다음과 같은 세션키 유도 함수를 이용하여 실제 세션키를 계산한다.

[세션키 유도 함수]

사용자와 웹 서버가 공유 비밀정보 C로부터 실제 세션키를 계산하는 세션키 유도함수의 자세한 동작 과정은 다음과 같다.

■ 입 력

- C : 비밀 공유 정보
- *keydatalen* : 세션키의 길이
- *hashlen* : 해쉬 함수의 출력 길이
- *SharedInfo* : 선택사항

■ 실행 과정

1) *counter* = 00000001(16진수)로 설정

2) For $I=1$ to $\lceil keydatalen/hahslen \rceil$

i) $Hash_i = H(Z \| counter \| [SharedInfo])$

ii) Increment $counter$

3) $keydatalen / hahslen$ = 정수이면,

$Hash! \lceil keydatalen/hahslen \rceil$

$= Hash \lceil keydatalen/hahslen \rceil$

정수가 아니면,

$Hash! \lceil keydatalen/hahslen \rceil$

$= Hash \lceil keydatalen/hahslen \rceil$

의 최상위

$(keydatalen - (hashlen \times \lfloor keydatalen/hahslen \rfloor))$비트

4) $SK = Hash1 \| Hash2 \| \ldots \| Hash \lceil keydatalen/hahslen - 1 \rceil \| Hash! \lceil keydatalen/hahslen \rceil$

■ 출 력

- 길이가 $keydatalen$인 세션키 SK

4. 제안하는 프로토콜 분석

4.1 제안하는 프로토콜의 특징

제안하는 프로토콜 1/2는 기존의 dhHybridOneFlow 프로토콜과 같이 양방향 묵시적 키 인증과 일방향 key freshness 제공한다. 그리고 두 방식 모두 proxy 서버는 일회용 키 쌍 생성에는 관여하지만, 사용자 A의 비밀키를 모르므로 사용자와 웹 서버 사이의 실제 세션키는 알 수 없다. 또한, 사용자 A에게 요구되는 모듈라 멱승 계산을 proxy 서버에게 의존함으로써, 단말 사용자에게 요구되는 계산량을 감소시킬 수 있으므로 무선 인터넷 환경의 휴대 단말기 등에 사용하기에 적합하다. 그리고 각 웹 서버에 대해 $y_B^{x_A}$는 고정된 값이므로, 사용자가 자주 사용하는 웹 서버에 대해 이 값을 사전에 계산하여 단말기에 저장해 두면, 세션키 생성에 필요한 계산량을 더욱 감소시킬 수 있다. 즉, 관용 암호 방식의 복호화 연산과 세션키 계산을 위한 해쉬 함수 연산만으로 사용자와 웹 서버는 매 세션마다 다른 세션키를 설정할 수 있게 된다.

제안하는 프로토콜 1에서는 proxy 서버가 전송하는 정보에 대해 디지털 서명을 생성함으로써, 웹 서버는 수신한 정보가 사용자 A의 요청에 의해 proxy 서버가 생성했음을 확인할 수 있다. 또한, 이를 통해 proxy 서버의 신원을 명시적으로 확인할 수 있다는 장점이 있다. 그러나, 이 방식은 서명 검증에 필요한 계산량으로 인해 웹 서버에게 5번의 모듈라 멱승이 요구된다는 단점이 있다. 이러한 문제점을 해결하기 위해, 웹 서버에게 요구되는 계산량을 기존의 dhHybridOneflow 방식과 동일하도록 개선한 것이 제안하는 프로토콜 2이다. 제안하는 프로토콜 2에서는, 웹 서버가 proxy 서버의 신원을 묵시적으로 인증 가능하며, 웹 서버에게 요구되는 계산량은 기존의 방식과 동일하다는 장점이 있다.

본 논문에서 제안하는 proxy 서버를 이용하는 변형 방식에서는 세션키를 생성하기 위해 모듈라 멱승 외에 관용 암호 방식의 복호화 과정이 한번씩 요구되지만, 이는 공개키 암호 관련 연산에 비해 계산량이 훨씬 작으므로 PDA나 핸드폰과 같은 무선 단말기에서도 연산이 가능하다.

4.2 안전성 분석

제안하는 프로토콜은 기존의 프로토콜에 비해 단말 사용자에게 요구되는 계산량을 감소시킨 방식이며, 기존의 양자간(two-party) 프로토콜의 안전성은 그대로 유지한다는 장점이 있다. 본 절에서는 제안하는 프로토콜의 안전성을 다음과 같이 나누어 분석한다[2, 3, 10].

1) 수동적 공격자에 대한 안전성

제안하는 프로토콜 1에서 proxy 서버를 제외한 다른 수동적 공격자가 공개 전송 정보를 이용하여 사용자 A와 웹 서버 B 사이의 세션키를 구하기 위해서는 사용자 A의 공개키 y_A와 웹 서버 B의 공개키 y_B로부터 $g^{x_A x_B} \bmod p$를 계산

〈표 1〉 제안하는 프로토콜과 기존 프로토콜의 비교

	dhHybridOneFlow	제안하는 프로토콜 1	제안하는 프로토콜 2
개체 인증	제공하지 않음	웹 서버는 proxy 서버 인증 가능	제공하지 않음
묵시적 키 인증	양방향	양방향	양방향
Key freshness	일방향	일방향	일방향
사용자 A의 계산량	모듈라 멱승 3번	모듈라 멱승 1번*	모듈라 멱승 1번*
웹 서버 B의 계산량	모듈라 멱승 3번	모듈라 멱승 5번	모듈라 멱승 3번

* 자주 사용하는 웹 서버에 대해서는 사전 계산하여 저장해 두는 것이 가능

해야 하고, $t_p = g^{r_p} \bmod p$와 y_B로부터 $g^{r_p x_B} \bmod p$를 계산해야 한다. 또한, 제안하는 프로토콜 2에서는 proxy 서버를 제외한 다른 수동적 공격자가 공개 전송 정보를 이용하여 사용자 A와 웹 서버 B 사이의 세션키를 구하기 위해서는, 사용자 A의 공개키 y_A와 웹 서버 B의 공개키 y_B로부터 $g^{x_A x_B} \bmod p$를 계산해야 하고 t_p'로부터 t_p 값을 계산해야 한다.

이것은 모두 Diffie-Hellman 문제와 일치하므로, 기존의 dhHybridOneFlow 프로토콜과 마찬가지로 제안하는 프로토콜 1/2에서 수동적 공격자가 세션키를 구하는 어려움은 Diffie-Hellman 문제의 어려움과 동치이다.

2) Proxy 서버에 대한 안전성

제안하는 프로토콜에서 proxy 서버는 세션키 생성에 필요한 정보의 일부를 계산하여 사용자 A에게 전송하지만, 사용자 A와 웹 서버 B 사이의 실제 세션키는 두 당사자만이 계산할 수 있어야 한다. 따라서, 실제 세션키 계산에 필요한 정보 중 일부는 사용자 A와 웹 서버 B 외에 다른 사용자는 계산할 수 없는 값으로 구성해야 한다.

제안하는 프로토콜 1/2에서 proxy 서버는 비밀 공유 정보 C를 생성하는데 필요한 $C' \equiv y_B^{r_p} \bmod p$ / $t_p = y_B^{r_p} \bmod p$ 값을 계산하여 사용자 A에게 전송하지만, 사용자 A의 비밀키 x_A를 알지 못하므로, 실제 C 값을 구하는 것은 계산상 불가능하다. 따라서, 제안하는 프로토콜에서 proxy 서버가 사용자와 웹 서버사이에 설정된 실제 세션키를 알아내는 것은 계산상 불가능하다.

3) 능동적 공격자에 대한 안전성

i) 공격자가 사용자 A로 위장하려는 경우(Active Impersonation)

제안하는 프로토콜 1/2에서 임의의 공격자가 사용자 A로 위장하여 proxy 서버에게 세션키 설정을 요구하는 경우, 사용자 A와 proxy 서버 사이에 사전에 공유한 비밀키를 알지 못하므로 proxy 서버로부터 받은 암호문을 복호할 수 없을 뿐만 아니라, 사용자 A의 비밀키를 모르므로 실제 세션키를 계산할 수 없다. 따라서, 제안하는 방식은 능동적 위장(active impersonation) 공격에 대해 안전하다.

ii) 사용자 A의 비밀키가 노출된 경우의 위장 공격 (Key compromised Impersonation)

제안하는 프로토콜 1/2에서 사용자 A의 비밀키가 노출되더라도, 이를 이용하여 사용자 A로 위장하려는 공격자는 사용자 A와 proxy 서버 사이에 사전에 공유한 비밀키를 알지 못하므로 C'를 복호할 수가 없다. 즉, 사용자 A의 비밀키가 노출된 경우에도 공격자는 실제 세션키를 구할 수 없게 된다.

또한, 제안하는 프로토콜 1에서는 사용자 A의 비밀키를 획득한 공격자가 웹 서버 B로 위장하려는 경우, proxy 서버가 매 세션마다 다른 랜덤 수를 이용하여 전송정보를 생성하면 웹 서버 B의 비밀키를 모르는 공격자는 현재 세션의 세션키를 계산할 수 없게 된다. 제안하는 프로토콜 2에서도, 웹 서버 B의 비밀키를 모르는 공격자는 t_p'로부터 t_p를 구하는 것이 계산상 불가능하다. 따라서, 기존의 dhHybridOneFlow 방식은 키 분배를 하고자 하는 두 사용자 중 한쪽만이 랜덤 수를 사용하므로, 사용자 A의 비밀키가 노출되는 경우 누구든지 사용자 A로 위장하여 세션키를 설정하는 공격이 가능하지만, 제안하는 프로토콜은 공격자가 사용자 A의 비밀키를 알게 되더라도 사용자 A로 위장할 수 없고, 사용자 A에게 웹 서버로 위장할 수도 없으므로 key-compromised resilience를 만족한다는 장점이 있다.

iii) Forward secrecy에 대한 안전성

제안하는 프로토콜 1에서는 세션키를 생성하기 위해 proxy 서버가 매 세션마다 다르게 선택한 랜덤수가 사용되므로 사용자 A의 비밀키 x_A가 노출되더라도 세션키의 안전성에는 아무런 영향을 미치지 않는다. 그러나 웹 서버 B의 비밀키 x_B가 노출되는 경우에는 누구든지 공개정보 y_A, t_p와 x_B를 이용하며 $C = y_A^{x_B} \bmod p \| t_p^{x_B} \bmod p$를 구할 수 있으므로 제안하는 프로토콜 1은 half forward secrecy를 만족한다. 제안하는 프로토콜 2에서도, 웹 서버 B의 비밀키 x_B가 노출되는 경우에는 누구든지 공개정보 y_{AP}, t_p'를 이용하여 $t_p = y_{AP}^{x_B} \bmod p \oplus t_p'$, $C = y_A^{x_B} \bmod p \| t_p$를 구할 수 있으므로 half forward secrecy만을 제공한다.

iv) Known Key Security

Known key security는 공격자가 프로토콜에 참여하지는 않고 노출된 과거의 세션키와 공개 정보를 이용하여 현재의 세션키를 구하려는 known key passive 공격과 현재 세션에 직접 참여하여 과거의 세션키와 공개 정보를 이용하여 정당한 사용자로 위장하여 세션키를 설정하는 known key impersonation 공격으로 나눌 수 있다. dhHybrid-OneFlow 프로토콜의 경우 공격자가 이전 세션에 사용자 A, B 사이에 설정한 세션키 C와 전송정보 t_p를 획득하는 경우, 후에 다시 공격자가 사용자 A로 위장하여 사용자 B에게 t_p를 전송함으로써 세션키를 설정하는 known key impersonation 공격이 가능하다. 그러나, 제안하는 프로토콜 1, 2는 세션키 생성에 매 세션마다 다르게 선택된 랜덤수가 사용되므로 공격자가 과거 세션키와 전송 정보를 획득하더라도 현재의 세션키를 구하는데 아무런 도움이 되지 않는다. 즉, 제안하는 프로토콜에서 공격자가 이전 세션키

〈표 2〉 제안하는 프로토콜의 안전성 분석 결과

	dhHybridOneFlow	제안하는 프로토콜 1	제안하는 프로토콜 2
Active Impersonation	안전함	안전함	안전함
Key Compromised Impersonation	안전하지 않음	안전함	안전함
Forward Secrecy	half FS 제공	half FS 제공	half FS 제공
Known Key Passive attack	안전함	안전함	안전함
Known Key Impersonation attack	안전하지 않음	안전함	안전함

C와 전송정보 t_p를 획득하고 사용자 A로 위장하여 세션키를 설정하려면 proxy 서버에게 세션키 설정을 요청해야 한다. 그러므로, 제안하는 프로토콜 1/2는 known key passive/impersonation 공격에 대해 안전하다. <표 2>는 제안하는 프로토콜과 기존의 프로토콜의 안전성을 분석한 결과를 정리한 것이다.

5. 결 론

무선 인터넷 접속 기술 및 무선 단말기의 발전으로 인해 휴대폰이나 PDA 등을 이용하는 무선 인터넷 서비스가 활성화되고 있다. 무선 인터넷 환경은 유선 환경에 비해 사용자들이 언제 어디서나 서비스를 이용할 수 있다는 장점이 있으므로, 앞으로 이를 이용한 서비스가 계속 증가할 것으로 생각된다.

또한, 2003년 하반기부터 무선 공개키 기반구조(Wireless Public Key Infrastructure) 서비스가 상용화될 예정이므로 이를 이용한 인터넷 뱅킹, 온라인 쇼핑, 주식 거래와 같은 다양한 무선 인터넷 서비스가 더욱 확대될 것이다.

보다 안전한 무선 인터넷 서비스를 제공하기 위해서는 유선 환경에서와 같이 암호 시스템의 사용이 필수적으로 요구되며, 암호 시스템의 사용에 있어 가장 중요한 요소가 키 분배 프로토콜이다. 그러나 지금까지 제안된 대부분의 키 분배 프로토콜은 공개키 암호 방식을 이용하므로 세션키 생성에 요구되는 계산량이 많다는 단점이 있다.

그러나, 무선 인터넷 환경은 유선 환경에 비해 사용하는 단말기의 계산 능력이나 전원 장치가 부족하므로, 유선 환경과 같이 공개키 암호 방식을 이용하는 키 분배 프로토콜을 적용하는데 많은 어려움이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해, proxy 서버를 이용하여 단말 사용자에게 요구되는 공개키 관련 연산을 입의의 서버에 의존하는 변형된 키 분배 프로토콜을 제안하였다. 제안하는 방식은 기존의 Diffie-Hellman 기반 방식의 프로토콜을 변형한 것으로, 무선 단말기 사용자들에게 요구되는 계산량은 감소시키면서 기존의 프로토콜이 갖는 안전성은 그대로 유지할 수 있다는 장점이 있다. 따라서, 제안하는 프로토콜은 상대적으로 적은 계산 능력을 갖는 무선 인터넷용 단말기에 적합한 방식으로 안전한 무선 인터넷 서비스의 활성화에 기여할 수 있을 것으로 기대된다.

참 고 문 헌

[1] ANSI X9.42, "Agreement of symmetric Key on Using Diffie-Hellman Cryptography," 2001.

[2] S. M. Ahn, B. Y. Song, M. Mambo, H. Shizuya, D. H. Won, "A study on the Security of Key Agreement Protocol based on Asymmetirc techniques against Passive attack," Symposium on Cryptography and Information Security(SCIS) 2002 IEICE, pp.955-960, 2002.

[3] S. Blake-Wilson, D. Johnson, A. Menezes, "Key agreement protocols and their security analysis," Cryptography and Coding, Lecture Notes in Computer Science 1355, pp.30-45, 1997.

[4] W. Diffie, M. E. Hellman, "New directions in cryptography," IEEE Transaction of Information Theory, IT 22, 6, pp.644-654, 1976.

[5] T. ElGamal, "A Public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory IT-31, pp.469-472, 1985.

[6] IEEE P1363, "Standard for Public-Key Cryptography," Working draft D13, 1999.

[7] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures : Delegation of the power to sign message," IEICE Trans, on Fundamentals, E79-A(9): 1338-1354, 1996.

[8] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures for delegating signing operation," Proc. Third ACM Conference on Computer and Communications

Security, pp.48-57, 1996.

[9] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997.

[10] S. H. Oh, K. J. Kim. K.A. Shim, M. Mambo and D. H. Won, "How secure are the Girault's Self-certified Public-key based Key agreement protocols against Active attacks?," Proc. of SCIS 2002, The 2002 Symposium on Cryptography and Information Security, Japan, pp.423-428, 2002.

양 형 규

e-mail : hkyang@kangnam.ac.kr

1983년 성균관대학교 전자공학과(공학사)

1985년 성균관대학교 대학원 전자공학과 (공학석사)

1995년 성균관대학교 대학원 정보공학과 (공학박사)

1984년~1991년 삼성전자 컴퓨터부문 선임연구원

1995년~현재 강남대학교 컴퓨터미디어공학부 부교수

관심분야 : 암호 프로토콜, 컴퓨터네트워크 보안